PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-215391
(43)Date of publication of application : 02.08.2002

(51)Int.Cl. G06F 9/44

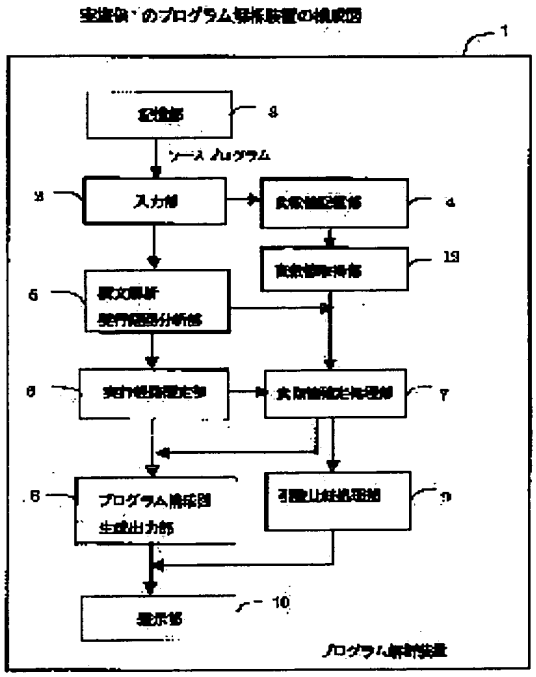
(21)Application number : 2001-345350
(22)Date of filing : 09.11.2001

(71)Applicant : FUJITSU LTD
(72)Inventor : MATSUO AKIHIKO
NAGAHASHI KENJI

(30)Priority
Priority number : 2000352622 Priority date : 20.11.2000 Priority country : JP

(54) DEVICE AND METHOD FOR PROGRAM ANALYSIS AND RECORDING MEDIUM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a device for extracting a subroutine by analyzing a program and accurately outputting the call relation thereof.
SOLUTION: This program analysis device, for generating and outputting the block diagram of a program predesignated based on the extracted subroutine comprises an execution route analysis means for analyzing the program and analyzing an execution route, a variable detection means for detecting the conditional variables of the conditional branch sentences, including a sub routine in the execution route and the call side variables of the subroutine from among the analyzed program, a variable value definition processing means for researching and defining the constant of the variable at a substitute side by tracing in the order, the path of the execution path of the program from a variable detection point, and an execution path limiting means for limiting the branch point of the conditional branch sentence, based on the defined variable value and extracting the sub routine defined by the call side from the limited path.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2002－215391
（P2002－215391A）
(43)公開日　平成14年８月２日(2002.8.2)

(51)Int.Cl.⁷	識別記号	ＦＩ	ﾃｰﾏｺｰﾄﾞ(参考)
Ｇ０６Ｆ　9/44		Ｇ０６Ｆ　9/06	６２０Ｌ　５Ｂ０７６

審査請求　未請求　請求項の数６　ＯＬ　（全 13 頁）

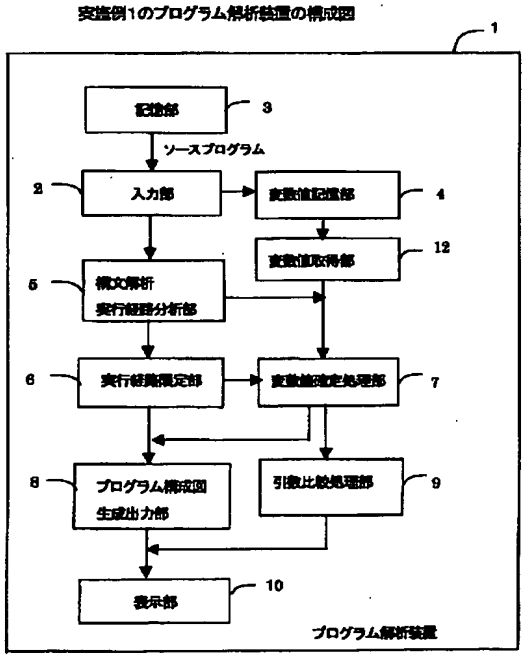
(21)出願番号　特願2001－345350(P2001－345350)
(22)出願日　平成13年11月９日(2001.11.9)
(31)優先権主張番号　特願2000－352622(P2000－352622)
(32)優先日　平成12年11月20日(2000.11.20)
(33)優先権主張国　日本（ＪＰ）

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(72)発明者　松尾　昭彦
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(72)発明者　長橋　賢児
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
(74)代理人　100108187
弁理士　横山　淳一

Ｆターム(参考)　5B076 DE04 EC02

(54)【発明の名称】　プログラム解析装置及びプログラム解析方法及び記録媒体

(57)【要約】
【課題】プログラムを解析してサブルーチンを抽出し、その呼出し関係を正確に出力する装置を提供すること。
【解決手段】プログラムを解析し、実行経路を分析する実行経路分析手段と、分析されたプログラムの中から実行経路にサブルーチンを含む条件分岐文の条件変数およびそのサブルーチンの呼出し先変数を検出する変数検出手段と、検出された変数について、変数検出箇所からプログラムの実行経路を順次遡り、変数の代入元の定数を探索し確定する変数値確定処理手段と、確定された変数値を基に条件分岐文の分岐箇所を限定し、限定した経路から呼出し先が確定したサブルーチンを抽出する実行経路限定手段とを備え、抽出したサブルーチンを基に予め指定されたプログラムの構成図を生成出力するプログラム解析装置。



1

【特許請求の範囲】

【請求項１】　予め指定されたプログラムから抽出されたサブルーチンを基にサブルーチンの中からサブルーチンを抽出する処理を繰返すサブルーチン抽出手段と、抽出されたサブルーチンを基に指定されたプログラムの構成図を生成し出力するプログラム構成図生成出力手段とを有するプログラム解析装置であって、

サブルーチン抽出手段は、プログラムを解析し、実行経路を分析する実行経路分析手段と、

分析されたプログラムの中から実行経路にサブルーチンを含む条件分岐文の条件変数およびそのサブルーチンの呼出し先変数を検出する変数検出手段と、

検出された変数について、変数検出箇所からプログラムの実行経路を順次遡り、変数の代入元の定数を探索し確定する変数値確定処理手段と、

確定された変数値を基に条件分岐文の分岐箇所を限定し、限定した経路から呼出し先が確定したサブルーチンを抽出する実行経路限定手段とを備えたことを特徴とするプログラム解析装置。

【請求項２】　変数値確定処理手段は、予めプログラムを実行させて取得した変数値を記憶する変数値記憶手段と、変数の代入元の定数を探索のときに、変数値記憶手段から変数に対応する変数値を取得する変数値取得手段とを備えたことを特徴とする請求項１記載のプログラム解析装置。

【請求項３】　分析された実行経路からサブルーチンを抽出し、抽出されたサブルーチンの使用変数の変数を確定する使用変数値確定手段と、

サブルーチン内の条件分岐文の各分岐箇所の条件変数の変数値を抽出する分岐箇所変数値抽出手段と、

使用変数値確定手段で得られた変数値と分岐箇所変数値抽出手段で抽出された変数値との比較内容を抽出し出力する変数値比較出力手段とを備えたことを特徴とする請求項１記載のプログラム解析装置。

【請求項４】　予め指定されたプログラムから抽出されたサブルーチンを基にサブルーチンの中からサブルーチンを抽出する処理を繰返すサブルーチン抽出ステップと、抽出されたサブルーチンを基に指定されたプログラムの構成図を生成し出力するプログラム構成図生成出力ステップとを有するプログラム解析方法であって、

サブルーチン抽出ステップは、プログラムを解析し、実行経路を分析する実行経路分析ステップと、

分析されたプログラムの中から実行経路にサブルーチンを含む条件分岐文の条件変数およびそのサブルーチンの呼出し先変数を検出する変数検出ステップと、

検出された変数について、変数検出箇所からプログラムの実行経路を順次遡り、変数の代入元の定数を探索し確定する変数値確定処理ステップと、

確定された変数値を基に条件分岐文の分岐箇所を限定し、限定した経路から呼出し先が確定したサブルーチンを抽出する実行経路限定ステップとを備えたことを特徴とするプログラム解析方法。

2

【請求項５】　予め指定されたプログラムから抽出されたサブルーチンを基にサブルーチンの中からサブルーチンを抽出する処理を繰返すサブルーチン抽出機能と、抽出されたサブルーチンを基に指定されたプログラムの構成図を生成し出力するプログラム構成図生成出力機能とを実現するプログラムを記録した記録媒体であって、

サブルーチン抽出機能は、プログラムを解析し、実行経路を分析する実行経路分析機能と、

分析されたプログラムの中から実行経路にサブルーチンを含む条件分岐文の条件変数およびそのサブルーチンの呼出し先変数を検出する変数検出機能と、

検出された変数について、変数検出箇所からプログラムの実行経路を順次遡り、変数の代入元の定数を探索し確定する変数値確定処理機能と、

確定された変数値を基に条件分岐文の分岐箇所を限定し、限定した経路から呼出し先が確定したサブルーチンを抽出する実行経路限定機能とを実現するプログラムを記録したコンピュータによって読取り可能な記録媒体。

【請求項６】　予め指定されたプログラムから抽出されたサブルーチンを基にサブルーチンの中からサブルーチンを抽出する処理を繰返すサブルーチン抽出機能と、抽出されたサブルーチンを基に指定されたプログラムの構成図を生成し出力するプログラム構成図生成出力機能とを実現させるためのプログラムであって、

プログラムを解析し、実行経路を分析する実行経路分析機能と、

分析されたプログラムの中から実行経路にサブルーチンを含む条件分岐文の条件変数およびそのサブルーチンの呼出し先変数を検出する変数検出機能と、

検出された変数について、変数検出箇所からプログラムの実行経路を順次遡り、変数の代入元の定数を探索し確定する変数値確定処理機能と、

確定された変数値を基に条件分岐文の分岐箇所を限定し、限定した経路から呼出し先が確定したサブルーチンを抽出する実行経路限定機能とを有するサブルーチン抽出機能を実現させるためのプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、複数のプログラムから構成される大規模なソフトウェアの作成および保守に不可欠な、処理内容の理解に用いられるサブルーチン呼び出し関係情報を生成出力する装置に関する。

【０００２】

【従来の技術】大規模なソフトウェアの構造を理解する際に不可欠と言えるのが、サブルーチンの呼び出し関係情報（サブルーチンによるプログラム構成図）である。大規模なシステムではサブルーチンが別のサブルーチンを呼んでいることがしばしばあり、注目しているプログ

ラムが実際にどのような処理を実行するのかをソースコードを追いながら把握するのは困難である。こうした場合に従来から、プログラムを構文解析し、サブルーチンの呼び出し関係を作成し表示する装置が用いられてきた。この装置を用いることで処理内容の把握が容易となり、また多くのプログラムの中からデータの読み込みを行う処理や書き出しを行う処理を行う箇所などを素早く特定することが可能になる。

【０００３】図12に、従来例のプログラム解析装置の構成図を示す。プログラム解析装置１は、入力部2 、記憶部３、構文解析部13、プログラム構成図生成出力部８、表示部10、サブルーチン抽出部11からなる。図13に、従来例のプログラム解析装置の処理の流れ図を示す。図14に従来のプログラムと表示例１を示す。

【０００４】図14(a) にプログラム例を、図14(b) に表示例を示す。まず、入力部２は、指定されたソースプログラムを記憶部３から読込み、その中の指定された対象プログラムを構文解析部13に渡す(S101 ｽﾃｯﾌﾟ)。次に、構文解析部13は、対象プログラムについて構文解析をする(S102 ｽﾃｯﾌﾟ)。 次に、サブルーチン抽出部11は、構文解析された対象プログラムの中からサブルーチン呼出し箇所をリストアップする(S103 ｽﾃｯﾌﾟ)。

【０００５】図14(a) のPGM01 、PGM02 に示す対象プログラムからは、サブルーチンDBIO001 、サブルーチンDBIO002 がリストアップされる。次に呼び出されるサブルーチンの中から更にサブルーチンをリストアップする(S104 ｽﾃｯﾌﾟ)。図14(a) に示すようにサブルーチンDBIO001 からは、サブルーチンDBWRITE,サブルーチンDBIO002 からは、DBREADが検出される。

【０００６】次に全サブルーチンが終了したか確認する(S105 ｽﾃｯﾌﾟ)。まだ、DBWRITE,DBREADについては、未チェックのため、S104ステップに戻る。 DBWRITE,DBREADのサブルーチンをリストアップする。DBWRITE,DBREAD内には、サブルーチンはないものとすると、何もリストアップされない。

【０００７】次に、全サブルーチン完了かを再度チェックする(S105 ｽﾃｯﾌﾟ)。全サブルーチン完了したため、その呼出し関係から呼出し関係情報すなわちプログラム構成図を生成し、表示部10に出力する(S106 ｽﾃｯﾌﾟ)。表示例を図14(b) に示す。

【０００８】

【発明が解決しようとする課題】大規模なソフトウェアでは、特定のデータファイルの読み書きなどの処理などもサブルーチン化され、システムの変更などによる影響箇所を少なくすることに貢献している。このようなアクセスサブルーチンは、データの読み込みや書き出しを一つのサブルーチンにまとめ、その引数の一つを制御コマンドとすることで記述を容易にしていることが多い。

【０００９】また、サブルーチンの呼び出し先を変数によって指定しておき実行時に値を確定させるような手法

も、処理効率や拡張性、柔軟性のためによく使われる。ところが、従来のプログラム解析装置では、呼び出し先のサブルーチンが変数の値によって決まる場合の呼び出し先を特定することが出来ず、充分な情報を表示することができなかった。例えば、上記のような制御コマンド形式のサブルーチンを呼び出している場合、その箇所からはデータの書き込みは行わないにも関わらず書き込みルーチンへの呼び出し関係が表示されてしまうといった問題があった。

【００１０】図15に従来例のプログラムと表示例２を示す。例えば、プログラムPGM03 では書き込み処理のためにDBIOSUB1を呼んでいるが、図15(b) に示すプログラム構成図上では、読み込み処理も行っているように表示されてしまう。また、プログラムPGM04 では読み込み処理のためにDBIOSUB2を呼んでいるが、図15(b) に示すプログラム構成図上では、書きこみ処理も行っているように表示されてしまう。

【００１１】また、このようなサブルーチンに対して、制御コマンドとして不適当な値を使っているため正常に動作しなかったり、使われることのない不要な制御コマンドが用意されていて性能上・保守上の問題になっているようなケースについても従来の技術ではチェックすることができず目視確認や動作テストが必要とされていた。

【００１２】本発明の目的は、プログラムを解析してサブルーチンを抽出し、その呼出し関係をより正確に出力するプログラム解析装置の提供にある。

【００１３】

【課題を解決するための手段】予め指定されたプログラムから抽出されたサブルーチンを基にサブルーチンの中からサブルーチンを抽出する処理を繰返すサブルーチン抽出手段と、抽出されたサブルーチンを基に指定されたプログラムの構成図を生成し出力するプログラム構成図生成出力手段とを有するプログラム解析装置であって、サブルーチン抽出手段は、プログラムを解析し、実行経路を分析する実行経路分析手段と、分析されたプログラムの中から実行経路にサブルーチンを含む条件分岐文の条件変数およびそのサブルーチンの呼出し先変数を検出する変数検出手段と、検出された変数について、変数検出箇所からプログラムの実行経路を順次遡り、変数の代入元の定数を探索し確定する変数値確定処理手段と、確定された変数値を基に条件分岐文の分岐箇所を限定し、限定した経路から呼出し先が確定したサブルーチンを抽出する実行経路限定手段とを備えた構成である。

【００１４】この構成により、ソースプログラムを読込み、まず分析対象プログラムのサブルーチンを抽出する。そしてサブルーチン内の実行経路上に、条件分岐文とサブルーチンを抽出し、条件分岐文の条件変数およびサブルーチンの呼出し先の変数を検出する。次に実行経路を順次遡りながら、その変数の代入元の定数を探索す

ることで、変数値を自動的に検出し確定する。そして、確定した定数により、条件分岐文の分岐場所が確定するので、その実行経路の呼出し先が確定したサブルーチンを抽出する。次に、このサブルーチンを呼出し、そのサブルーチン内についても、これらの処理を繰り返すことで、サブルーチンの呼び出し関係が確定する。そして分析対象プログラムから順次呼出されるサブルーチンの経路が決まるので、この経路のサブルーチンからなるプログラム構成図を生成する。この結果、より正確なプログラム構成図を出力することが可能となる。

【００１５】また、変数値確定処理手段は、予めプログラムを実行させて取得した変数値を記憶する変数値記憶手段と、変数の代入元の定数を探索のときに、変数値記憶手段から変数に対応する変数値を取得する変数値取得手段とを備えた構成である。この構成により、予めプログラムの実行により取得した変数値を使用することができるのでプログラムの実行経路が明確となり、より正確なプログラム構成図を生成出力することが可能となる。

【００１６】また、分析された実行経路からサブルーチンを抽出し、抽出されたサブルーチンの使用変数の変数を確定する使用変数値確定手段と、サブルーチン内の条件分岐文の各分岐箇所の条件変数の変数値を抽出する分岐箇所変数値抽出手段と、使用変数値確定手段で得られた変数値と分岐箇所変数値抽出手段で抽出された変数値との比較内容を抽出し、出力する変数値比較出力手段とを備えた構成である。

【００１７】この構成により、予めサブルーチンを呼び出す側のサブルーチンの変数値の抽出と、呼出される側のサブルーチンの条件分岐文の分岐箇所の変数値との比較ができる。この結果、不一致により、エラーを発見し、修正できるので、より正確なサブルーチンの呼び出し関係から生成されるプログラム構成図をより正確に出力することが可能となる。

【００１８】

【発明の実施の形態】図１にプログラム解析装置の実施例１の構成図を示す。プログラム解析装置１は、入力部２、記憶部３、変数値記憶部４、構文解析実行経路分析部５、実行経路限定部６、変数値確定処理部７、プログラム構成図生成出力部８、引数比較処理部９、表示部10、変数値取得部12から構成される。

【００１９】図２に実施例１のプログラム解析装置の処理の流れ図を示す。図６に実施例のプログラムと表示例を示す。図６(a) の分析対象プログラムPGM03,PGM04 に対する実行処理について、説明を行う。まずソースプログラム名、対象プログラム名を入力部２により指定する。そして入力部２は、指定されたソースプログラムを記憶部３から読み込み、分析対象プログラム名ともに構文解析実行経路分析部５に渡す。

【００２０】図６(a) の例では、対象プログラムとしてPGM03 ,PGM04が指定される。(S11ｽﾃｯﾌﾟ)。次に、構文解析実行経路分析部５は、対象プログラムPGM03 に対して、構文を解析し実行経路分析を行う(S12ｽﾃｯﾌﾟ)。分析結果から、プログラムPGM03 の条件分岐箇所をリストアップする(S13ｽﾃｯﾌﾟ)。プログラムPGM03 には、条件分岐箇所はないため、S15ｽﾃｯﾌﾟへ進む。

【００２１】次に実行可能性のあるサブルーチンの呼出し箇所をリストアップする。サブルーチン'DBIOSUB1'が検出される(S15ｽﾃｯﾌﾟ)。図４に実施例１の変数値確定処理の流れ図を示す。次にサブルーチンの呼出し先の変数を変数値確定処理部７にて確認する(S16ｽﾃｯﾌﾟ)。サブルーチンの呼出し先は、変数ではなく、固定値 'DBIOSUB1' であるので、変数値確定済として何もせずに終了する(S31ｽﾃｯﾌﾟ)。

【００２２】次に呼び出されるサブルーチンの処理をすべて完了しているか確認する(S17ｽﾃｯﾌﾟ)。サブルーチン'DBIOSUB1'の処理が未了のため、対象プログラムにDBIOSUB1を設定し、また、S12 ｽﾃｯﾌﾟに戻り、同様な処理を繰り返す。構文解析実行経路分析部５は、対象プログラムDBIOSUB1に対して、構文を解析し実行経路分析を行う(S12ｽﾃｯﾌﾟ)。

【００２３】分析結果から、サブルーチンDBIOSUB1のプログラムの中から実行経路にサブルーチンを含む条件分岐箇所をリストアップする(S13ｽﾃｯﾌﾟ)。図６(a) では、IF文３箇所（図６(a) の❸～❺）が検出される。次に条件分岐箇所について、実行経路限定部６にて実行経路限定処理を行う(S14ｽﾃｯﾌﾟ)。

【００２４】図３に実施例１の実行経路限定処理の流れ図を示す。分岐箇所で使用されている変数をリストアップする(S21ｽﾃｯﾌﾟ)。CMDID がリストアップされる。変数について、変数値確定処理部７に値を問い合わせる(S22ｽﾃｯﾌﾟ)。まず変数値が確定しているか否かをチェックする(S31ｽﾃｯﾌﾟ)。

【００２５】まだ、未確定のため、サブルーチンDBIOSUB1の条件分岐箇所である注目箇所（図６(a) の❸）の前に、実行される文があるか否かを判定する(S32ｽﾃｯﾌﾟ)。実行文を呼出し元のプログラムPGM03 まで戻り、直前の文（図６(a) に❶）をリストアップする。(S33ｽﾃｯﾌﾟ)。その文に対して、確定実行処理を呼び出す(S34ｽﾃｯﾌﾟ)。

【００２６】図５に実施例１の確定実行処理の流れ図を示す。その文は注目変数CMDID の値を変更しているかを確認する(S41ｽﾃｯﾌﾟ)。変更していないため、注目個所をその文にして、変数値確定処理に戻る(S46ｽﾃｯﾌﾟ)。まず変数値が確定しているか否かを確定値リストでチェックする(S31ｽﾃｯﾌﾟ)。　まだ、未確定のため、サブルーチンDBIOSUB1の呼出し箇所である注目箇所（図６(a) の❶）の前に、実行される文があるか否かを判定する(S32ｽﾃｯﾌﾟ)。

【００２７】実行文があるため、その直前の文（図６(a) に❷）をリストアップする。(S33ｽﾃｯﾌﾟ)。その文

に対して、確定実行処理を呼び出す(S34ｽﾃｯﾌﾟ)。その文は注目変数CMDID の値を変更しているかを確認する(S41ｽﾃｯﾌﾟ)。注目変数CMIDは'WT'で変更されている。

【００２８】次に、その文が代入している値が何かを確認する(S42ｽﾃｯﾌﾟ)。'WT'は、定数のため、確定値リストに入れる(S43ｽﾃｯﾌﾟ)。そして、変数値確認処理に戻る。変数値処理で、すべての変数値が完了したかを確認する(S30ｽﾃｯﾌﾟ)。すべての変数値が完了したため、図３に戻る。

【００２９】変数が確定しているので、それを取得する(S23ｽﾃｯﾌﾟ)。次に確定値 'WT' により、分岐条件をチェックする(S24ｽﾃｯﾌﾟ)。IF文の中で、'WT'が成立する箇所（図６（a)の③）があるため、その成立時以外経路、図６(a) の④及び⑤は、実行可能性なしとする(S25ｽﾃｯﾌﾟ)。次に、図２に戻ると、IF文の中で、'WT'が成立する箇所の実行可能性のあるサブルーチン'DBWRITE'（図６(a) の(1))を抽出する（S15 ｽﾃｯﾌﾟ）。

【００３０】サブルーチンの呼出し先は、変数でなく固定値'DBWRITE' のため、変数値確定処理で、何もせずに戻る（S16 ｽﾃｯﾌﾟ）。次に呼び出されるサブルーチンの処理をすべて完了しているか確認する(S17ｽﾃｯﾌﾟ)。DBWRITE の処理が未了のため、対象プログラムにDBWRITE を設定し、また、S12 ｽﾃｯﾌﾟに戻り、同様な処理を繰り返す。

【００３１】図示はしていないが、DBWRITE に条件分岐、サブルーチンを含まないものとする。DBWRITE について、構文解析、実行経路解析が行われて(S12ｽﾃｯﾌﾟ)、条件分岐およびサブルーチンがないので、S17 ｽﾃｯﾌﾟに行き、全サブルーチンを対象プログラムとして処理完了のため、処理を終了する。

【００３２】この結果、プログラムPGM03 の中から、サブルーチンDBIOSUB1が抽出され、更にサブルーチンDBIOSUB1の中からサブルーチンDBWRITE が抽出されるプログラムの実行経路が確定する。次に、全指定プログラムが完了したかをチェックする（S18 ｽﾃｯﾌﾟ）。未了のため、プログラムPGM04 についても同様の処理を行う。

【００３３】その結果、プログラムPGM04 の中から、サブルーチンDBIOSUB1が抽出され、更にサブルーチンDBIOSUB1の中からサブルーチンDBREADが抽出されるプログラムの実行経路が確定する。そして全ての指定プログラムが完了すると、プログラム構成図生成出力部８は、プログラムのサブルーチンの実行経路について得た情報に基いてプログラム構成図を生成し、表示部に出力する（S19 ｽﾃｯﾌﾟ）。図６に表示部の出力例を示す。　また、変数値の確定を上記のように、自動で行わずに、画面上に変数値問い合わせのメッセージを表示させ、オペレータがマニュアルで指定する方法もある。　また、対象となるプログラム中の注目箇所に、その箇所での変数の値を外部に記録するプログラムを追加し、その対象となるプログラムをあらかじめ動作させておくことで変数の値の記録を作成しておく。そして変数値取得部12でこの記録を参照する。

【００３４】図７に実施例１の変数値記憶処理の説明図を示す。変数の値を外部に記録する処理を追加した例である。二重線の枠で囲まれた部分が追加したプログラムである。このプログラムPGM10 は変数CMDID を使って指定されるプログラムDBIOSUB3を呼び出しているが、CMDID の値はファイルから読み込まれるためプログラム解析だけでは値が決められない。そのため、この変数の内容を、ログファイルに出力するサブルーチンを追加することで記録している。

【００３５】すなわち変数CMDID の引数となるCMD をCMD-FILEから読み出したときにログをとり、変数値記憶部４に格納している。ログ記録完了後に、プログラム解析装置１にて、図５の確定実行処理を行うときに、S42 ｽﾃｯﾌﾟで、その文が代入している値CMD を確認したときに（図７の②）、通常変数のため、注目変数CMIDをその変数CMD にして（S45 ｽﾃｯﾌﾟ）変数値確定処理を再帰的に呼出す（S46 ｽﾃｯﾌﾟ）。

【００３６】そして変数値確認処理で、変数値未確認のため、注目処理の前の文（図７の③）をリストアップし、そして再度、図５の確定実行処理を行う。その文は、注目変数CMD を変更している（S41 ｽﾃｯﾌﾟ）。そのため、その文が代入している値は、外部入力（図７の③）のため、変数値取得部12は、ログファイルから変数CMD に対応する定数を呼出してくる。この値を確定値リストに設定することで、変数値確定処理が完了する。

【００３７】また、図８に実行経路の強調表示例を示す。プログラム構成図生成出力部８では、呼出し関係を表示するときに、サブルーチン名ではなく、指定プログラムから呼び出されるサブルーチンのソースを、実行される可能性がある範囲を例えばハイライトにより区別して表示することでより見やすい形態となる。この表示例を図８に示す。この図は、第６（ａ）のプログラムPGM04 の例であり、CMDID='RD'のときの実行経路を強調表示している。

【００３８】次に、図９に実施例２の処理の流れ図を示す。まずソースプログラム名、対象プログラム名を入力部２により指定する。そして入力部２は、指定されたソースプログラムを記憶部３から読み込み、分析対象プログラム名ともに構文解析実行経路分析部５に渡す(S51ｽﾃｯﾌﾟ)。次に全対象プログラムを構文解析実行経路分析部５にて構文解析、実行経路分析を行う(S52ｽﾃｯﾌﾟ)。

【００３９】次に、引数比較処理部９は、サブルーチンを抽出し、そのサブルーチンの引数の一覧表を作成する(S53ｽﾃｯﾌﾟ)。図10に実施例２のプログラム例を示す。図11に実施例２のサブルーチン引数比較図の説明図を示す。図10のプログラムPGM03,PGM04,PGM07 の場合は、サブルーチンDBIOSUB1が共通で抽出される。

【００４０】そして、変数値確定処理部７にて、使用引

数の変数値を確定する。変数CMDIDに対して、PGM03は、引数'WT'、PGM04は、引数'RD'、PGM07は、引数'RE'を検出する。また、変数DENPYOに対して、PGM03は、引数'不定' PGM04は、引数'不定' PGM07は、引数'不定'を検出する。次に、抽出した引数をもとに、引数比較処理部９は、この一覧表を生成する。

【００４１】次に、指定プログラムから呼出されるサブルーチンDBIOSUB1について、構文解析、実行経路分析を行う(S54ｽﾃｯﾌﾟ)。引数比較処理部９は、条件分岐箇所について使用変数を抽出する(S55ｽﾃｯﾌﾟ)。　すなわち、DBIOSUB1のIF文の分岐箇所の使用変数CMDIDに対応する変数値である引数'WT','RD','CL'を検出する。そして、引数値一覧表と比較する。

【００４２】プログラムPGM07から呼び出すサブルーチンDBIOSUB1の変数CMDIDの引数'RE'に対応するものがサブルーチンDBIOSUB1の中のプログラムにない。また、サブルーチンDBIOSUB1の変数CMDIDの引数'CL'に対応するものを使用して呼出す指定プログラムが見つからない。そして、これらの比較した結果を表示部10に表示出力する(S56ｽﾃｯﾌﾟ)。

【００４３】

【発明の効果】本発明では、サブルーチンの呼び出し箇所における変数の値によって呼び出し先が変わったり処理内容が大きく異なるような場合においても呼び出し先に関する正確な情報を表示できるため、プログラムの作成や保守に必要な処理内容の把握が容易となり、その結果作業に必要な工数が削減される。

【図面の簡単な説明】

【図１】　実施例１のプログラム解析装置の構成図

【図２】　実施例１のプログラム解析装置の処理の流れ図

＊【図３】　実施例１の実行経路限定処理の流れ図

【図４】　実施例１の変数値確定処理の流れ図

【図５】　実施例１の確定実行処理の流れ図

【図６】　実施例のプログラムと表示例

【図７】　実施例１の変数値記録処理の説明図

【図８】　実施例１の実行経路の強調表示例の説明図

【図９】　実施例２のプログラム解析装置の処理の流れ図

【図１０】　実施例２のプログラム例

【図１１】　実施例２のサブルーチンの引数比較図の説明図

【図１２】　従来例のプログラム解析装置の構成図

【図１３】　従来例のプログラム解析装置の処理の流れ図

【図１４】　従来例のプログラム例と表示例１

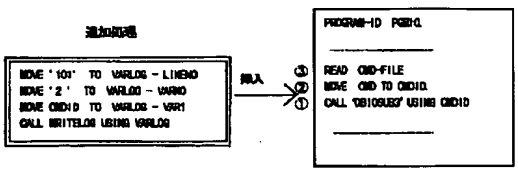
【図１５】　従来例のプログラム例と表示例２

【符号の説明】

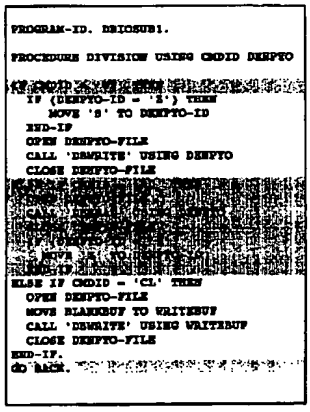
１　プログラム解析装置
２　入力部
３　記憶部
４　変数値記憶部
５　構文解析実行経路分析部
６　実行経路限定部
７　変数値確定処理部
８　プログラム構成図生成出力部
９　引数比較処理部
１０　表示部
１１　サブルーチン抽出部
１２　変数値取得部
１３　構文解析部

＊

【図７】

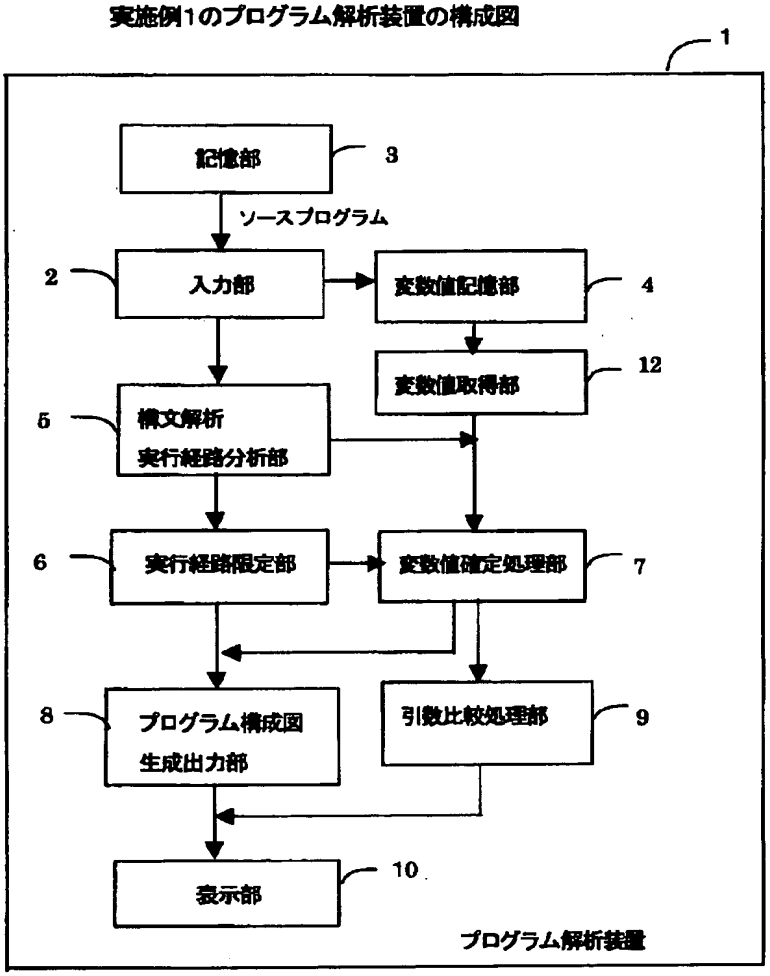


【図８】



【図１】

実施例1のプログラム解析装置の構成図



【図１１】

実施例2のサブルーチン引数比較図の説明図

(a) サブルーチンの引数一覧表

サブルーチン名	呼出し元	第一引数	第二引数
DBIOSUB1	PGM03	WT	不定
	PGM04	RD	不定
	PGM07	RE	不定

(b) サブルーチン内の条件分岐の引数比較

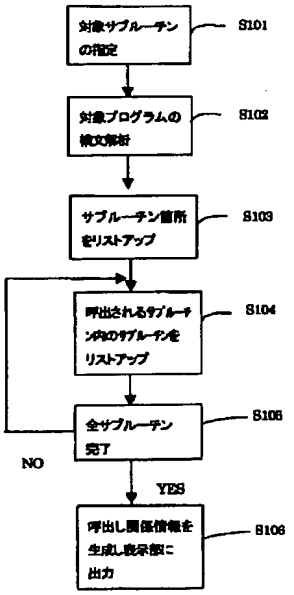
サブルーチン DBIOSUB1
第一引数に与えられる定数値 'RD', 'RE', 'WT'
第一引数と比較している定数値 'WT', 'RD', 'CL'
第二引数に与えられる定数値 なし
第二引数と比較している定数値 なし

警告(第一引数が'RE'の場合を判定する処理が含まれません
(呼び出し元 PGM07)

警告(第一引数を'CL'として呼び出すプログラムが有りません

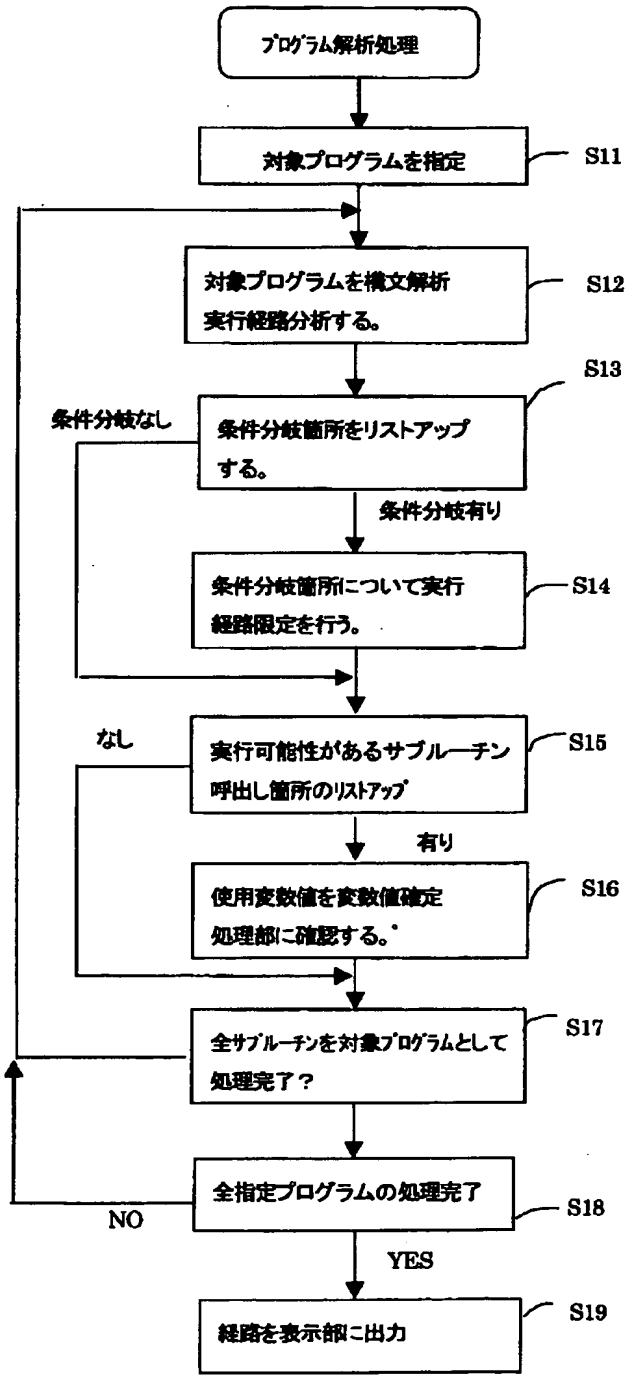
【図１３】

従来例のプログラム解析装置の処理の流れ図

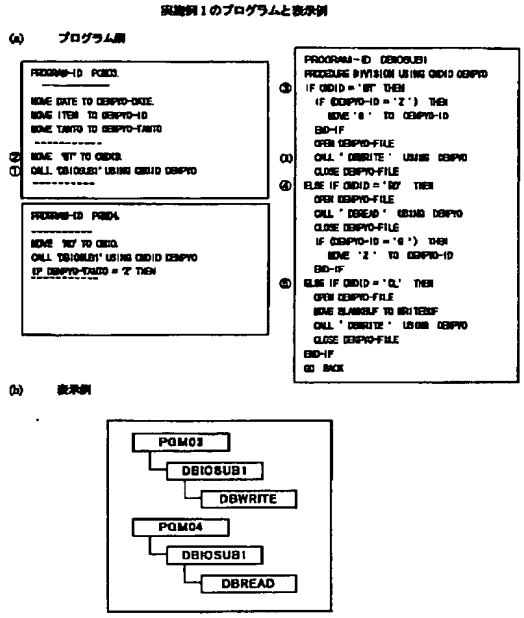


【図２】

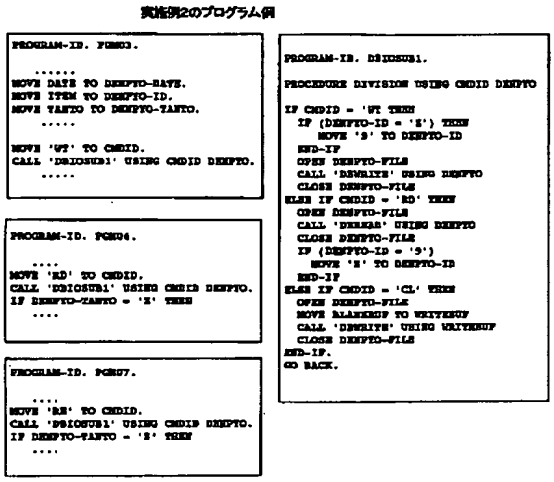
実施例1のプログラム解析装置の処理の流れ図



【図６】

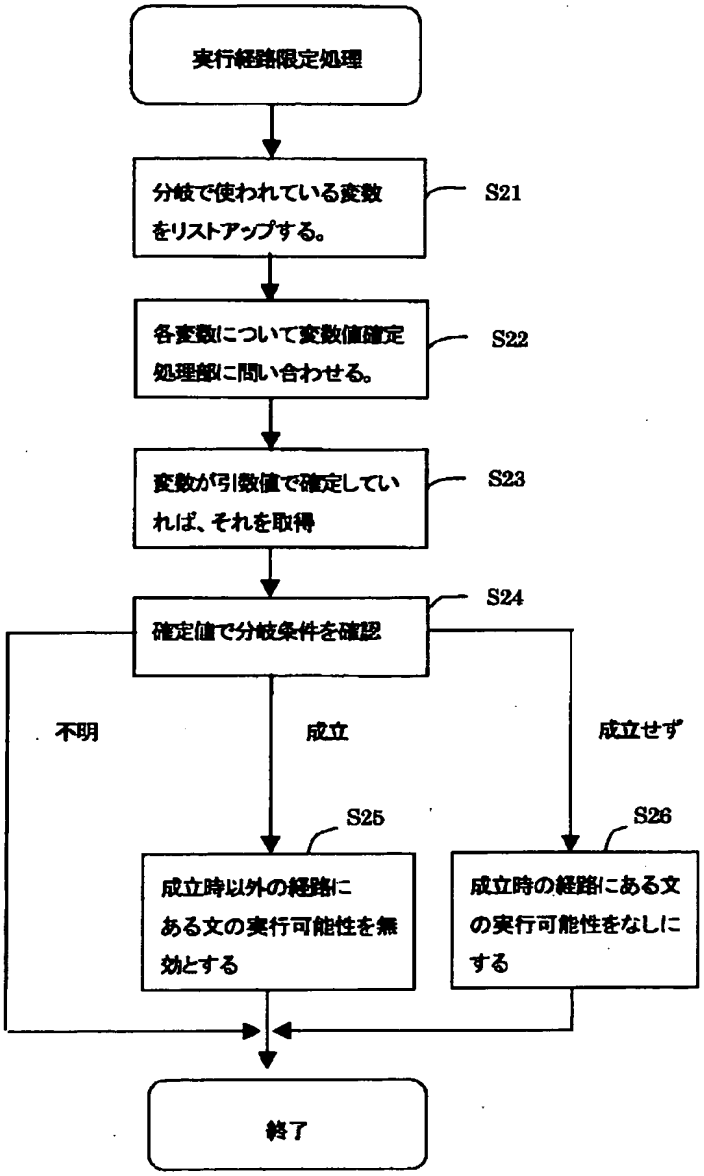


【図１０】



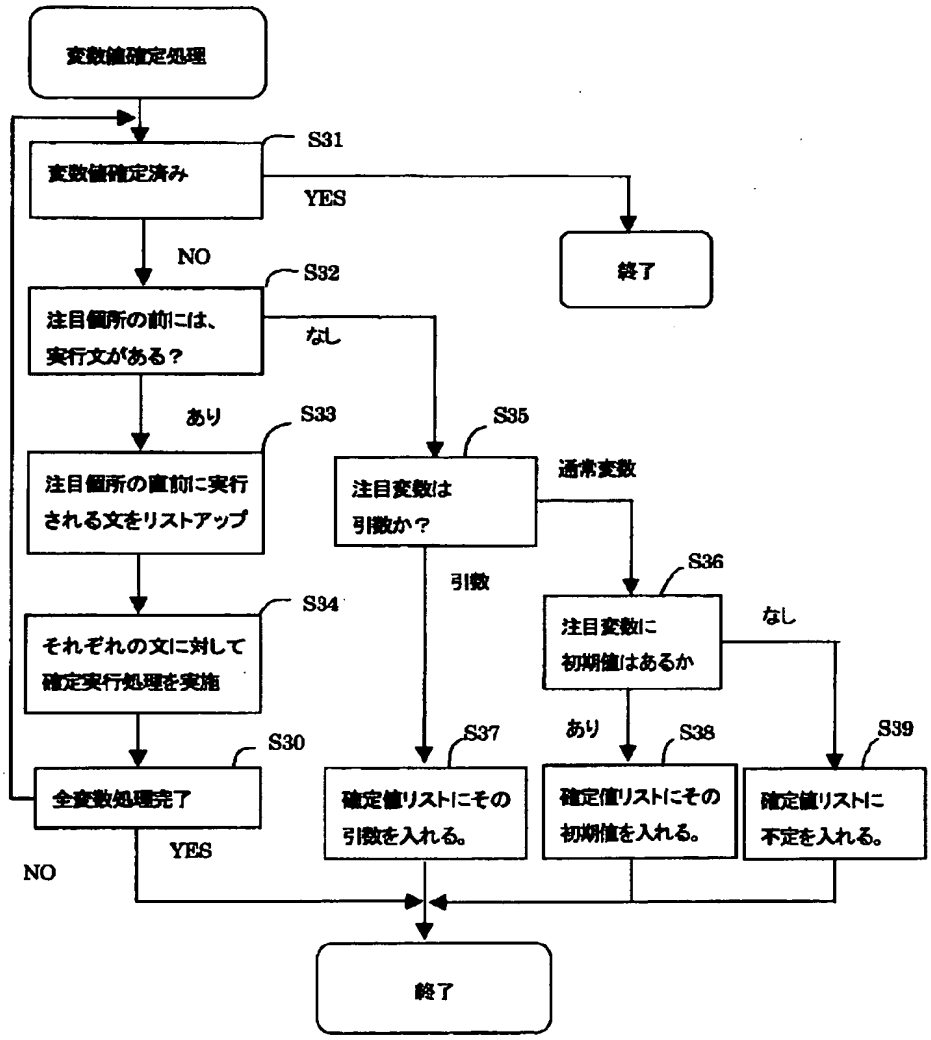
【図３】

実施例1の実行経路限定処理の流れ図



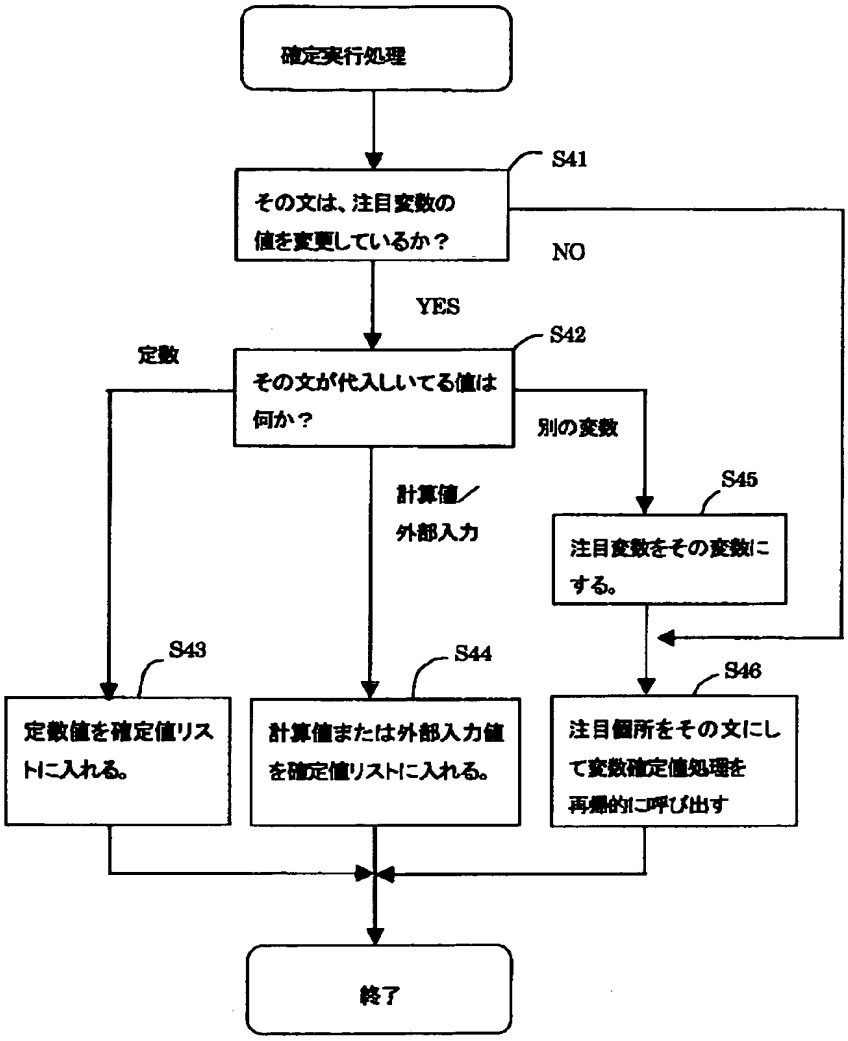
【図４】

実施例1の変数値確定処理の流れ図



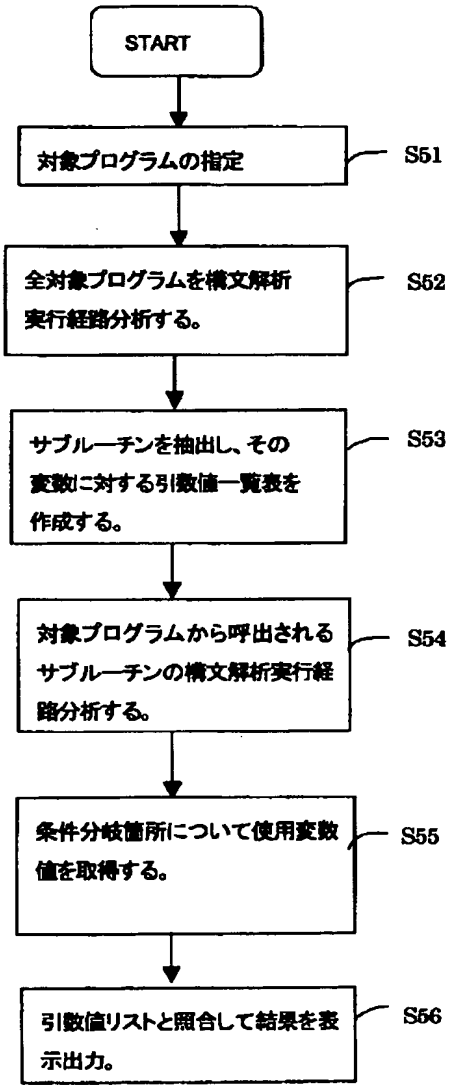
【図５】

実施例１の確定実行処理の流れ図

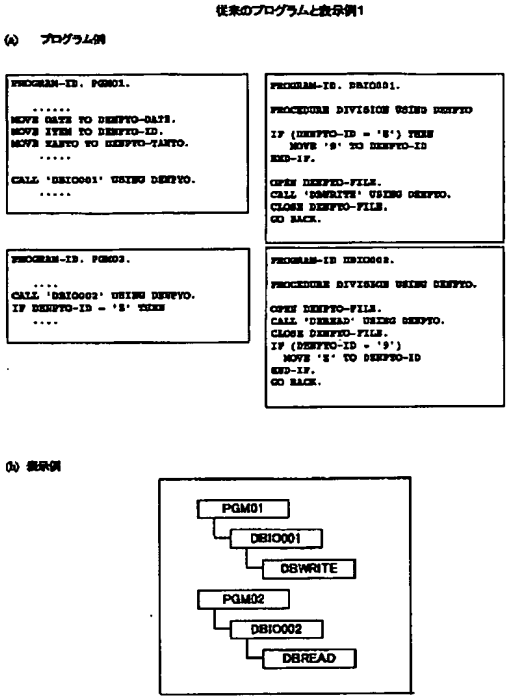


【図９】

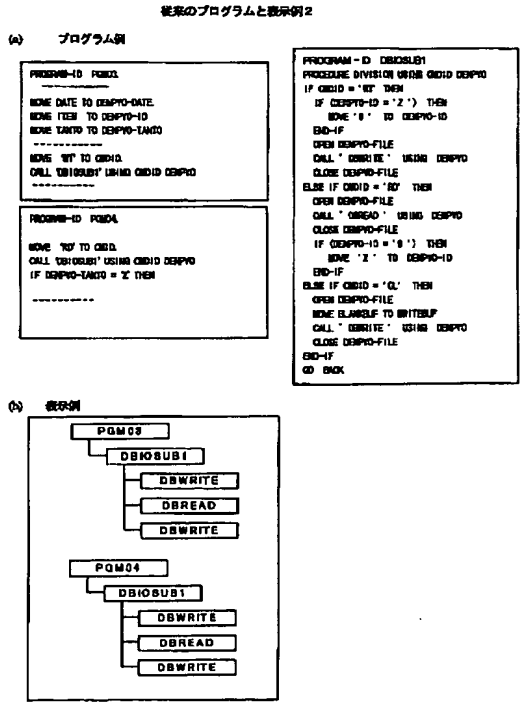
実施例2のプログラム解析装置の処理の流れ図



【図１４】



【図１５】



【図１２】

従来例のプログラム解析装置の構成図